ADVANCED WEB 2.0 PROGRAMMING TECHNICAL GUIDE by STEVEN OWINO

Deploy Complete Responsive Multi-Vendor, Online Ecommerce Platform built with CodeIgniter Framework

Support of following features

1. Multi-Vendor
2. MultiLanguage
3. Virtual products support
4. Multi Templates support
5. API
6. Administration with high level of access
7. Ajax based shopping cart
8. Checkouts are saved to administration, email notifications for every new order
9. Quantity of products managed from orders
10. Add textual pages
11. Activate and disable pages
12. File manager in administration
13. Integrated blog
14. Email subscribe
15. Easy installation
16. Readable source
17. Complete editing of public texts
18. Receive payments, also have option for paypal sandbox testing
19. Fast-loading templates for good seo
20. Site color change with easy to use gradient generator
21. Add multilanguage cookie notificator from silktide.com for EU Cookie Law
22. Multiple templates.. also can easily create your own..
23. Bank account payments support
24. Highcharts statistics for orders
25. Discount codes
26. Available on English
27. Responsive public pages, responsive administration, responsive vendor pages
28. Advanced search with treeView categories
29. Easy manage of products (new filed for every added language, subcategories are easy to manage)
30. Advanced sorting and order products

PREAMBLE
We MUST constantly keep in mind that Differentials are absolutely nothing, and we can conclude nothing from them,
except that their mutual ratios reduce to finite quantities. [Leonard Euler]

Prerequisites: Mind to understand Object Oriented Programming[All Ages]

Competent administrative skills for Windows and Linux Operating Systems e.g configuring Windows Hosted Network

Its a lot WORK!!!!! If you dont get it to work, learn what you dont know[study and practice], then it will WORK.

First, install SUCCESSFULLY Apache Ofbiz and understand its Framework.
Then.....After 10,000 Hours of patience(practice):

Install Xampp Server and place the "Ecommerce-CodeIgniter-Bootstrap-master" folder in Xampp's "htdocs" directory.

Login to administration
Users registration

```
### Reference Artifacts : Follow this to fully understand the design of the
SITE[Ecommerce-CodeIgniter-Bootstrap-master]
httpd.conf         :  Main Apache HTTP server configuration file/directives giving
the server instructions
my.ini                 :  MySQL configuration file for small systems
C:\Windows\System32\Drivers\etc\hosts: Host file for Microsoft TCP/IP for Windows.
Contains the mappings of the IP addreses to host
names
database.php      :  Settings needed to access your database
config.php           :  URL to your CodeIgniter root/base URL, WITH a trailing
slash
php.ini                  :  Configuring many of the aspects of PHP's
behavior/Enables Gzip output compression for faster page loads
.htaccess              :  Sets index folder, enables the rewrite engine for
loading css/js and sets the root path to specified Environment i.e
production
index.php             :  Loads the application environment, error reporting,
systemm folder, application folder, view folder names including
default  controller, custom   configuration values and resolves(sets the system
directory correctly) the  system path for
reliabilty, sets main path constants and finally loads the bootstrap file.
routes.php            : Re-map URI requests to specific controller
functions/methods/class
CodeIgniter.php   : This is the application development framework for PHP/system
initialization file(loads the base classes and executes the
                                 requests), sets security procudure on globals
,subclass prefix in the  config file, loads the Composer autoloader,
                                 starts the timer, instantiates the hook class,
config class(importantly loaded first), loads charset encoding,
                                loads compatibility features, instantiates the UTF-
8, URI, routing and output classes, validates a cache file,
                    loads the security file for xss and carf support, loads the
input class and sanitize globals, loads the language class,
                   app controller and local controller, sets a mark point for
benchmarking(time and base classes),
                   sanity checks(for empty class names, query strings  not going
through file_exists() check,
                   regular request for a non-existing page) handled as 404 error,
checks which method will be executed as a constructor,
                  global overrides, checks if $e404 flag is reset(sets the
rsegments starting from index, if there is a pre_controller hooker,
                  instantiates the requested controller, if there is a
"post_controller_constructor" hook, call the requested method,
                 if there is a "post-controller hook", and sends the final RENDERED
OUTPUT TO THE BROWSER,
                 checks if there is a "post_system hook".
constants.php     : Loads the framework constants
common.php      : Loads the global functions
autoload.php      : Specifies which systems should be loaded by default
mbstring.php      : PHP ext/mbstring compatibility package
hash.php               : PHP ext/hash compatibility package
password.php      : PHP ext/standard/password compatibility package
standard.php       : PHP ext/standard compatibility package
controller.php      : This class object is the super class that every library in
CodeIgniter will be assigned to.
 settings.php(controller), footer.php(admin_parts), bootstrap-select.min.js,
bootstrap.min.js, bootbox.js, zxcvbin.js,zxcvbin_bootstrap3.js,
 pGenerator.jquery.js, mine_admin.js, Home_admin_model.php, header.php,
home.php(admin_views), highcharts.js, data.js, drilldown.js,
 settings.php(modules), bootstrap-toggle.min.css, ckeditor.js,
 REST_Controller.php(A fully RESTful server implementation for CodeIgniter using
one library, one config file and one controller),
```

controller.php(third party), base.php(This library extends the CodeIgniter CI_Controller class and creates an application object allowing use of the HMVC design pattern),
language.php , products.php, orders.php, home.php(templates), footer.php(templates), bootstrap-confirmation.min.js, placeholders.min.js,
bootstrap-datepicker.min.js, contacts.php, shopping_cart.php(templates), Home.php(controller), Public_model.php, loop.php,
Format.php(convert between various formats such as XML, JSON, CSV, etc), ShoppingCart.php(class for manage products/libraries),
Home.php(admin_controller), Checkout.php(controller), checkout.php(templates), custom.css(templates), login.php, (templates),
header.php(template), view_product.php(template), image-preveiw.js , bootstrap.css(normalize), font-awesome.min.css

Common Technical Issues to be Solved

1. Apache Server running on port 80 for Windows(httpd.conf) and services.
2. Subnetting [Accessing your site from a domain other than the "localhost"] hosts file
3. Registering and hosting your sites Domain for global access.
4. Monetary transactions[checkout]. and [FINALLY],
5. Add User Login and Registration in the "Redlabel" template[This is your ultimate simple test if you have mastered object oriented design].

REFERENCES

Apache Ofbiz Development(Rupert Howell), Gradle in Action(Benjamin Muschko), Groovy in Action(Dierk Konig),
PHP Manual, Core PHP Programming(Leon Atkinson), Learning PHP5(David Sklar) Object Oriented Programming with PHP5(Hasin Hayder)
http://www.undersunconsulting.com/
https://apache-ofbiz.org/
https://code-projects.org/Ecommerce-site-in-PHP-with-source-code[and 73 video tutorials in youtube For Practice]
https://code-projects.org/Ecommerce-CodeIgniter-Bootstrap-master[MAIN SOURCE FILE FOR YOUR PRODUCTION SITE]

###CREDITS
http://gmmbsolutions.com/
https://github.com/kirilkirkov

###Developer
Steven Owino
brakets1990@protonmail.com
+254 772 328 507
+254 790 164 331